

## MULTIPROCESSOR SYSTEM AND TRANSACTION CONTROL METHOD THEREOF

Publication number: JP2001216259

Also published as:

Publication date: 2001-08-10

US2001013080 (A)

Inventor: KAMEYAMA SHIN; AKASHI HIDEYA; UEHARA KEITARO; TSUSHIMA YUJI; HAMANAKA NAOKI

Applicant: HITACHI LTD

Classification:

- international: G06F13/38; G06F13/38; (IPC1-7): G06F13/42; G06F13/38; G06F15/177

- european: G06F13/38A4

Application number: JP20000032738 20000204

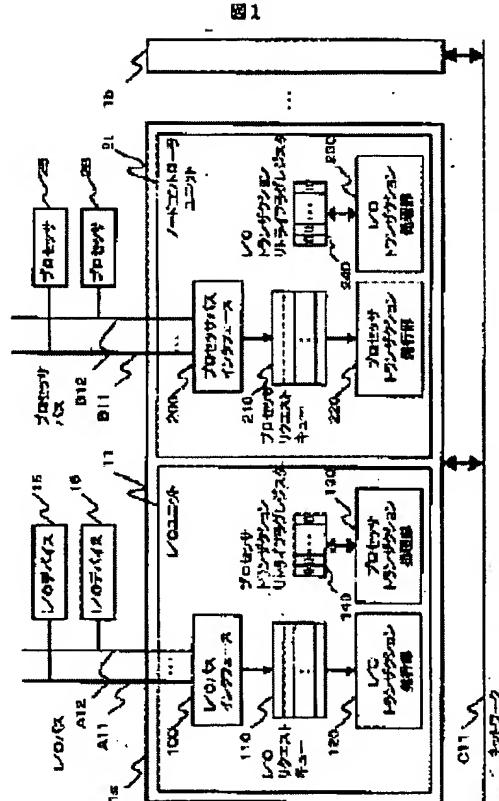
Priority number(s): JP20000032738 20000204

[Report a data error here](#)

### Abstract of JP2001216259

**PROBLEM TO BE SOLVED:** To make successively issuable a following transaction without waiting for the completion of a preceding transaction by achieving order guarantee and to accelerate the I/O bus transfer speed.

**SOLUTION:** The successive transactions of a posted write system are transferred among nodes 1a-1b. When one of the series of the transactions whose request origin is a device on the I/O buses A11-A12 of 1a is turned to retry in a request destination node, the following transaction is also turned to the retry and the issuance part 120 of a request origin node reissues the transactions. When 1a is the request destination node, a received transaction is processed in a processing part 230. When it can not be processed, however, a retry flag is raised at the bit corresponding to a generation source I/O bus of a retry flag register 240 and the retry is returned to the request origin. The retry is returned at the time of receiving the following transaction whose generation source is the same I/O bus as well.



Data supplied from the [esp@cenet](mailto:esp@cenet) database - Worldwide

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-216259

(P2001-216259A)

(43)公開日 平成13年8月10日 (2001.8.10)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 13/42  
13/38  
15/177

識別記号  
3 1 0  
3 4 0  
6 7 6

F I  
G 0 6 F 13/42  
13/38  
15/177

テマコード<sup>8</sup>(参考)  
3 1 0 5 B 0 4 5  
3 4 0 D 5 B 0 7 7  
6 7 6 A

審査請求 未請求 請求項の数14 O L (全 13 頁)

(21)出願番号 特願2000-32738(P2000-32738)

(22)出願日 平成12年2月4日 (2000.2.4)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 亀山 伸

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 明石 英也

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74)代理人 100075096

弁理士 作田 康夫

最終頁に続く

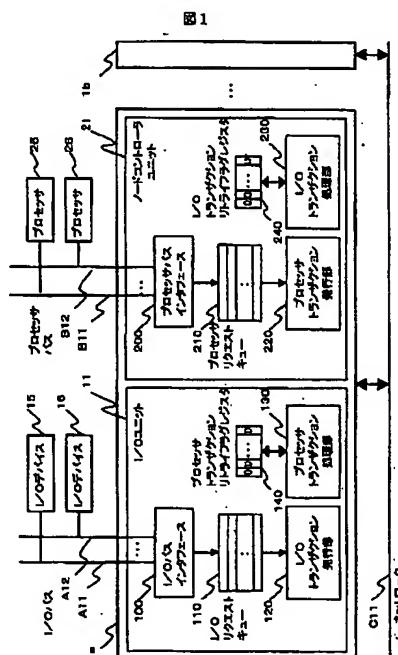
(54)【発明の名称】 マルチプロセッサシステム及びそのトランザクション制御方法

(57)【要約】 (修正有)

【課題】順序保証を達成することで先行トランザクションの完了を待たずに後続トランザクションを連続発行することを可能とし、I/Oバス転送を高速化する。

【解決手段】ボステップドライト方式の連続したトランザクションが複数ノード1a・1b間で転送される。

1aのI/OバスA11~A12上のデバイスが要求元である一連のトランザクションのうちの1つが要求先ノードでリトライとされば、後続するトランザクションもリトライとされ、要求元ノードの発行部120はこれらトランザクションを再発行する。1aが要求先ノードであれば、受信トランザクションを処理部230で処理するが、処理不可能ならリトライフラグレジスタ240の発生源I/Oバスに対応するビットにリトライフラグを立てて要求元にリトライを返送し、同一I/Oバスを発生源とする後続トランザクションの受信時にもリトライを返送する。



## 【特許請求の範囲】

【請求項1】 I/Oデバイスが接続される少なくとも1つのI/Oバスが各々結合された複数のI/Oユニットと、プロセッサユニットが接続される少なくとも1つのプロセッサバスが各々結合される複数のノードコントローラユニットとがネットワークで相互結合されたマルチプロセッサシステムにおいて、各I/Oユニットには、結合されたI/Oバスから受け付けた一連のI/O発トランザクションを、先行するI/O発トランザクションの要求先からの確認応答の受信に関わりなく要求先のノードコントローラユニットに向けて順次発行し、かつ発行済みのI/O発トランザクションについて要求先からリトライが要求されたときには再発行するI/Oトランザクション発行部を備え、各ノードコントローラユニットには、発行されたI/O発トランザクションの要求先が自ユニットに結合されたプロセッサバスであるときトランザクション処理を行うI/Oトランザクション処理部を備え、各I/Oトランザクション処理部は、I/O発トランザクションを受信して処理が不可能と判定すると要求元のI/Oユニットにリトライの要求を返送し、かつリトライの要求を返送したトランザクションの発生源であるI/Oバスと同一のI/Oバスを発生源とする後続するI/Oトランザクションについては処理を実施せずに要求元のI/Oユニットにリトライの要求を返送することを特徴とするマルチプロセッサシステム。

【請求項2】 I/Oデバイスが接続される少なくとも1つのI/Oバスが各々結合された複数のI/Oユニットと、プロセッサユニットが接続される少なくとも1つのプロセッサバスが各々結合される複数のノードコントローラユニットとがネットワークで相互結合されたマルチプロセッサシステムにおいて、

前記各ノードコントローラユニットは、結合されたプロセッサバスから受け付けた一連のプロセッサ発トランザクションを、先行するプロセッサ発トランザクションの要求先からの確認応答の受信に関わりなく要求先のI/Oユニットに向けて順次発行し、かつ発行済みのプロセッサトランザクションについて要求先からリトライが要求されたときには再発行するプロセッサトランザクション発行部を備え、

前記各I/Oユニットには、発行されたプロセッサ発トランザクションの要求先が自ユニットに結合されたI/Oバスであるときトランザクション処理を行うプロセッサトランザクション処理部を備え、

各プロセッサトランザクション処理部は、プロセッサ発トランザクションを受信して処理が不可能と判定すると要求元のノードコントローラユニットにリトライの要求を返送し、かつリトライの要求を返送したトランザクションの発生源であるプロセッサバスと同一のプロセッサバスを発生源とする後続するプロセッサ発トランザクション

については処理を実施せずに要求元のプロセッサバスにリトライの要求を返送するマルチプロセッサシステム。

【請求項3】 I/Oデバイスが接続される少なくとも1つのI/Oバスが各々結合された複数のI/Oユニットと、少なくとも1つのプロセッサユニットが各々結合される複数のノードコントローラユニットとがネットワークで相互結合されたマルチプロセッサシステムにおいて、

10 前記各ノードコントローラユニットは、結合されたプロセッサユニットから受け付けた一連のプロセッサ発トランザクションを、先行するプロセッサトランザクションの要求先からの確認応答の受信に関わりなく要求先のI/Oユニットに向けて順次発行し、かつ発行済みのプロセッサ発トランザクションについて要求先からリトライが要求されたときには再発行するプロセッサトランザクション発行部を備え、

前記各I/Oユニットには、発行されたプロセッサ発トランザクションの要求先が自ユニットに結合されたI/Oバスであるときトランザクション処理を行うプロセッサトランザクション処理部を備え、

各プロセッサトランザクション処理部は、プロセッサ発トランザクションを受信して処理が不可能と判定すると要求元のノードコントローラユニットにリトライの要求を返送し、かつリトライの要求を返送したトランザクションの発生源であるプロセッサユニットと同一のプロセッサユニットを発生源とする後続するプロセッサ発トランザクションについては処理を実施せずに要求元のプロセッサユニットにリトライの要求を返送するマルチプロセッサシステム。

【請求項4】 I/Oデバイスが接続される少なくとも1つのI/Oバスとのインタフェースを各々が有する複数のI/Oユニットと、プロセッサユニットが接続される少なくとも1つのプロセッサバスとのインタフェースを各々が有する複数のノードコントローラユニットとがネットワークで相互結合されたマルチプロセッサシステムであり、前記複数のノードコントローラユニットもしくは前記複数のI/Oユニットの任意のものが要求元ユニットとして他のユニットを要求先とするトランザクションを発行する機能を有するマルチプロセッサシステムにおいて、要求先となったユニットが1つのトランザクションについて要求元のユニットにリトライの要求を返送した場合に、該要求先のユニットは前記1つのトランザクションの発生源と同一のバスが発生源である後続するトランザクションについてもリトライの要求を返送し、前記要求元ユニットはリトライが要求されたトランザクションを再発行することを特徴とするマルチプロセッサシステムのトランザクション制御方法。

【請求項5】 前記要求先ユニットは、一旦リトライの要求を返送した後は、受信したトランザクションが再発行

されるトランザクションのうちの先頭のもの場合に限り処理可能か否かを判定してリトライ要求か処理実行かを再決定することを特徴とする請求項4記載のマルチプロセッサシステムのトランザクション制御方法。

【請求項6】I/Oデバイスが接続される少なくとも1つのI/Oバスとのインターフェースを各々が有する複数のI/Oユニットと、プロセッサユニットが接続される少なくとも1つのプロセッサバスとのインターフェースを各々が有する複数のノードコントローラユニットとがネットワークで相互結合されたマルチプロセッサシステムであり、前記複数のノードコントローラユニットもしくは前記複数のI/Oユニットの任意のものが要求元ユニットとして他のユニットを要求先とするトランザクションを発行する機能を有するマルチプロセッサシステムにおいて、前記要求元ユニットは、1つのバスからの一連のトランザクションのうちの1つのトランザクションについてリトライの要求を受けると、該1つのトランザクション、及び該1つのトランザクションの発生源と同一バスを発生源とする発行済みのトランザクションの全てを順次再発行し、前記要求先となったユニットは、1つのトランザクションについて処理不可能と判断してリトライを返送すると、該1つのトランザクションの発生源と同一バスを発生源とする後続するトランザクションを受信しても処理を実施しないことを特徴とするマルチプロセッサシステムのトランザクション制御方法。

【請求項7】アクセス要求元となるデバイスが接続される複数の第1種のバスが個別に、もしくは複数ごとに結合される複数の第1のユニットと、  
アクセス要求先となるデバイスが接続される複数の第2種のバスが個別に、もしくは複数ごとに結合される複数の第2のユニットと、  
前記複数の第1のユニット、複数の第2のユニットを相互結合するネットワークとを有し、  
前記第1のユニットの各々には、結合された第1種のバスを発生源とする一連のトランザクションを、先行するトランザクションの要求先からの確認応答の受信にかかりなく要求先に向けて順次発行し、かつ発行済みのトランザクションについて要求先である前記第2のユニットからリトライが要求されたときには再発行するトランザクション発行手段を有し、

前記第2種のユニットの各々には、発行された前記トランザクションを要求先として受信した場合、処理が不可能の時には要求元である前記第1のユニットにリトライの要求を返送し、かつリトライの要求を返送したトランザクションの発生源である前記第1種のバスと同一の前記第1種のバスを発生源とする後続するトランザクションに対しても、前記要求元である第1のユニットにリトライの要求を返送するトランザクション処理手段を有するトランザクション制御システム。

【請求項8】前記第2のユニットの各々には、前記第1

種のバスの数に対応したエントリーを有し、各エントリーには対応する前記第1種のバスを発生源とするトランザクションについてリトライを要求したことを示すリトライフラグが記録されるレジスタを有する請求項6記載のトランザクション制御システム。

【請求項9】前記レジスタの各エントリーのリトライフラグは、対応する前記第1種のバスを発生源とするトランザクションのうち再発行されたものの先頭が受信され、当該第2のユニットでトランザクション処理が可能と判定されたときクリアされることを特徴とする請求項8記載のトランザクション制御システム。

【請求項10】前記トランザクション処理手段は、前記リトライフラグが記録されている期間中に対応する前記第1種のバスを発生源とする後続のトランザクションを受信したら該後続のトランザクションについても要求元の前記第1種のユニットにリトライの要求を返送することを特徴とする請求項9記載のトランザクション制御システム。

【請求項11】前記トランザクション発行手段は、再発行する一連のトランザクションの先頭のトランザクションにはヘッダフラグを附加して再発行することを特徴とする請求項7記載のトランザクション制御システム。

【請求項12】前記第1のユニットの各々には、結合された第1種のバスから入力するトランザクションを順番に格納するトランザクションキューを有し、該トランザクションキューにはトランザクションが当該ユニットから始めて発行するトランザクションか、リトライ要求により再発行するトランザクションかを区別する情報を格納するトランザクション属性フィールドを有する請求項7記載のトランザクション制御システム。

【請求項13】前記第1のユニットの各々には、結合された第1種のバスから入力するトランザクションを順番に格納するトランザクションキューを有し、該トランザクションキューにはトランザクションが当該ユニットから始めて発行するトランザクションか、リトライ要求により再発行するトランザクションかを区別する情報を格納するトランザクション属性フィールドを有し、前記トランザクション発行手段は、同一の第1種のバスからの一連のトランザクションのうちの再発行するものの先頭であるトランザクションを前記トランザクションキュー属性フィールドの値から判別し、再発行の際にヘッダフラグを附加することを特徴とする請求項7記載のトランザクション制御システム。

【請求項14】アクセス要求元となるデバイスが接続される複数の第1種のバスが個別に、もしくは複数ごとに結合される複数の第1のユニットと、

アクセス要求先となるデバイスが接続される複数の第2種のバスが個別に、もしくは複数ごとに結合される複数の第2のユニットと、

前記複数の第1のユニット、複数の第2のユニットを相

互結合するネットワークとを有し、前記第1のユニットの各々には、結合された第1種のバスを発生源とする一連のトランザクションを、先行するトランザクションの要求先からの確認応答の受信にかかりなく要求先に向けて順次発行し、1つのトランザクションについて要求先である前記第2のユニットからリトライが要求されたときには、前記一連のトランザクションのうち前記1つのトランザクションから後続して既に発行したトランザクションの全てを順次再発行するトランザクション発行手段を有し、

前記第2種のユニット各々には、発行されたトランザクションを要求先として受信したとき処理可能か否かを判定し、処理が不可能の時には要求元である前記第1のユニットにリトライの要求を返送し、かつリトライの要求を返送したトランザクションの発生源である前記第1種のバスと同一の前記第1種のバスを発生源とする後続するトランザクションを受信しても処理の実行を行わないトランザクション処理手段を有するトランザクション制御システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、I/Oバス上のデバイスを要求元または要求先とするトランザクションの順序を保証したマルチプロセッサシステム、もしくはより一般に、トランザクション制御システムに関する。

【0002】

【従来の技術】I/Oバスまたはプロセッサバスが発行するライトトランザクションをシステムで処理する方法の一種にポステッドライトがある。ポステッドライトはライトトランザクションをシステムが受け付けた直後に要求元バスで完了させ、以後の処理はシステムが責任を持って行う処理方法である。ライトトランザクションが発行されると、当該トランザクションがシステム上で完了する前にバスまたはプロセッサから後続トランザクションが発行可能になり、性能が向上する。

【0003】

【発明が解決しようとする課題】しかしながらPCIバス仕様レビジョン2、1などの規定ではPCIバス上のI/Oデバイスを要求元にしたライトトランザクションは順序を保証しなければならないという制約がある。依存関係のあるトランザクションを連続して発行した際に、先行のトランザクションがリトライされた場合には、後続のトランザクションが先に完了すると順序が保証されなくなる。ここでリトライとは要求先が一時的にアクセスを受け付けられない状態にある場合などに、要求元に対してしばらく後にあらためてアクセスを実行するよう要求することである。したがって、あるPCIバスがライトトランザクションをシステム側に発行すると、当該ライトがリトライされずに完了が保証されるまで当該PCIバスは後続トランザクションをシステム側

に発行しないことでこの制約を守っている。その結果、PCIバスから依存関係のあるトランザクションが連続して発行された場合には性能が低下してしまうという問題点があった。

【0004】同様にプロセッサを要求元とし、PCIバス上のI/Oデバイスを要求先にしたライトトランザクションにおいても、順序を保証する必要があるため、先行のライトトランザクションが完了するまで後続のライトトランザクションが連続発行できずにスループットが低下する問題があった。

【0005】本発明は上記の従来問題点を鑑み、依存関係のあるトランザクションの順序保証をしつつ要求元が先行のライトトランザクションの完了を待たずに後続トランザクションを連続発行することでバストランザクションを高速化することを目的とする。

【0006】

【課題を解決するための手段】I/Oデバイスが接続される少なくとも1つのI/Oバスが各々結合された複数のI/Oユニットと、プロセッサユニットが接続される少なくとも1つのプロセッサバスが各々結合された複数のノードコントローラユニットとがネットワークで相互結合されたマルチプロセッサシステムにおいて、本発明は以下の構成により前記従来の技術の課題を解決する。

【0007】まずI/Oバスを発生源としてノードコントローラに結合されるプロセッサバスに向けられるI/O発トランザクションの制御に関しては、トランザクションの要求元となる各I/Oユニットに、結合されたI/Oバスから受け付けた一連のI/O発トランザクションを、先行するI/O発トランザクションの要求先からの確認応答の受信にかかりなく要求先のノードコントローラユニットに向けて順次発行し、かつ発行済みのI/O発トランザクションについて要求先からリトライが要求されたときには再発行するI/Oトランザクション発行部を備える。一方、要求先となる各ノードコントローラユニットにはI/Oトランザクション処理部を備え、各I/Oトランザクション処理部は、I/O発トランザクションを受信して処理が不可能と判定すると要求元のI/Oユニットにリトライの要求を返送し、かつリトライの要求を返送したトランザクションの発生源であるI/Oバスと同一のI/Oバスを発生源とする後続するI/Oトランザクションについては処理を実施せずに自動的に要求元のI/Oユニットにリトライの要求を返送する。

【0008】このように、要求先ユニットにて、同一バスを発生源とする一連のトランザクションの1つをでリトライとしたら、それに後続するトランザクションもリトライとする構成により、後から発行したトランザクションが要求先ユニットで先行するトランザクションよりも先に処理が受け付けられ、処理順序が逆転する事態が防止される。つまり、一連トランザクションの順序保証

がなされ、要求元ユニットにて要求先からトランザクション完了の確認応答の返送を待って次のトランザクションを発行するという発行方法をとる必要がなくなり、高速のトランザクションが可能となる。

【0009】より具体的には、要求先となるユニット、上記の構成では各ノードコントローラユニットに、トランザクションの発生源である個々のI/Oバスに対応したエントリーを有するリトライフラグレジスタを設け、その各エントリーには対応するI/Oバスを発生源とするトランザクションについてリトライを要求したらそのことを示すリトライフラグを記録する。これを利用して、同一I/Oバスからの一連のトランザクションの一つを一旦リトライとすれば後続のトランザクションもリトライとすることができます。

【0010】要求先ユニットにてリトライを解消する、つまり対応するリトライフラグをクリアするのは、リトライ要求により再発行される一連のトランザクションのうち先頭のもの（最初にリトライとされ、再発行されたトランザクション）を受信し、要求先ユニットにて処理可能であった時である。再発行している一連トランザクションのうちの後続のものが受信されても、自動的にリトライを繰り返すのみである。この、処理可能かの判定を行い、可能ならばリトライを解消し、トランザクション処理に移行するのか、それともリトライを自動的に繰り返すか、の区別のため、要求元となる各I/Oユニットのトランザクション発行部は、再発行である一連のトランザクションの先頭にヘッダフラグを付して再発行する。1つのI/Oユニットに複数のI/Oバスが結合される構成では、ヘッダフラグはそのうち個々のI/Oバスを発生源とする一連のトランザクションの先頭をも意味するようにしても良い。さらに、各I/Oユニットのトランザクションキューニーには、始めて発行するトランザクションか再発行するトランザクションかを区別して記録するためのトランザクション属性フィールドを設ける。

【0011】以上にI/Oバスを発生源としてノードコントローラに結合されたプロセッサバスに向けられるI/O発トランザクションの制御に関する本発明の特徴的構成についてのべたが、プロセッサバスを発生源としてI/Oバスに向けられるプロセッサ発トランザクションについても全く同様の構成が取られる。即ち、要求元となる各ノードコントローラユニットに、プロセッサバスから受け付けた一連のトランザクションを要求先のI/Oユニットに向けて連続して発行し、かつ発行済みのトランザクションについて要求先からリトライが要求されたときには再発行するトランザクション発行部を備え、一方、要求先となる各I/Oユニットにはプロセッサトランザクション処理部を備え、各プロセッサトランザクション処理部は、トランザクション処理が不可能と判定すると要求元のノードコントローラユニットにリトライ

の要求を返送し、かつリトライの要求を返送したトランザクションの発生源であるプロセッサバスと同一のプロセッサバスを発生源とする後続するトランザクションについては自動的にリトライとする。

【0012】要求先となる各I/Oユニットにリトライフラグレジスタを設ける点、若しくは要求元となる各ノードコントローラユニットのトランザクション発行で再発行するトランザクションの先頭のものにヘッダフラグを付す点などの具体的構成も、先に述べたI/Oトランザクションの制御のための構成と同様である。

【0013】要求先からの1つのトランザクションに対するリトライの返送に対応して、要求元では既に送信済みの一連のトランザクション全てを連続して再発行するよう構成する变形も可能である。この变形の場合、要求先では、一旦1つのトランザクションにリトライを返送した場合、後続トランザクションについてリトライを返送する必要はなくなる。ただし、後続トランザクションについていは、受信しても処理を行わない。このような構成では返送の通信オーバヘッドが小さくなり、より高速のトランザクション処理が可能となる。

【0014】なお、上記ではトランザクションの管理単位を単一のプロセッサバス単位としているが、単一のプロセッサユニット単位でも良い。I/Oトランザクションに関しては、発行されるI/Oトランザクションの最終的な要求先が各プロセッサユニットとなる。また、プロセッサトランザクションに関しては、各プロセッサトランザクション処理部は、リトライの要求を返送したトランザクションの発生源であるプロセッサユニットと同一のプロセッサユニットを発生源とする後続するトランザクションについて自動的にリトライとする構成となる。各I/Oユニットのリトライフラグレジスタは、発生源であるプロセッサユニット単位にリトライフラグが設定する構成となる。

【0015】先頭トランザクションを判別するためにトランザクションにヘッダを付与する代わりにヘッダフラグをトランザクションにエンコードする構成とすることも可能である。その他の本発明の特徴は実施態様の説明にて明らかにされる。

【0016】

【発明の実施の形態】以下、I/Oバス上のデバイスを要求元としたトランザクションとしてI/O発メモリライドを例に、またI/Oバス上のデバイスを要求先としたトランザクションとしてプロセッサバス発I/Oライドを例として、本発明の実施例を図面を参照しつつ説明する。以降の説明では、I/O発のトランザクションをI/Oトランザクション、またプロセッサバス発のトランザクションをプロセッサトランザクションと記す。また一連のトランザクションと記した場合には依存関係のあるトランザクション、より詳しくは単一のバス（I/Oバス、もしくはプロセッサバス）を発生源とする連続

する一群のトランザクションのことを意味している。

【0017】図1は本発明の一実施例に係るマルチプロセッサシステムの構成を示す概略図である。図1において、ノード1aに含まれるI/Oユニット11には複数のA11～A12が接続され、これらを介して複数のI/Oデバイスが15, 16が接続される。なお、図では個々のI/Oバスにそろぞれ1つずつのI/Oデバイスが接続されるが、各I/Oバスに複数のI/Oデバイスを接続する構成も可能である。一方、ノードコントローラユニット21には複数のプロセッサバスB11～B12が接続され、これらを介して複数のプロセッサユニット25, 26が接続可能である。ここでも、各プロセッサバスに複数のプロセッサユニットを接続する構成も可能である。I/Oユニット11はI/Oバス上のデバイスから発行されたトランザクションまたはI/Oバス上のデバイスに対して発行されたトランザクションの制御を行う。またノードコントローラユニット21はプロセッサバス上のプロセッサユニットから発行されたトランザクションまたはプロセッサバス上のプロセッサユニットに対して発行されたトランザクションの制御を行う。ノード1bに示すとおり、上記と同様な構成のノードが複数個、多重に設けられ、それらのノード間がネットワークC11によって結合されてマルチプロセッサシステムを構成している。なお、I/Oユニットかノードコントローラユニットかのいずれか一方だけを備えたノードが、図示した双方を備えたノードと混在するマルチプロセッサシステムも可能である。また、I/Oユニットだけのノードとノードコントローラだけのユニットでマルチプロセッサシステムを構成することもできる。ネットワークC11は内部に図示しないバッファを持ち、各ノードから発行されたトランザクションをバッファリングさせる。このため、各ノードからのトランザクションの連続発行が可能である。

【0018】I/Oユニット11は複数のI/OバスA11～A12とのI/Oバスインタフェース100、I/OバスA11～A12からのトランザクションを格納するI/Oリクエストキューリストキュー110、I/Oリクエストキューリストキュー110に格納されたトランザクションを先行のトランザクションの完了を待たずに順次発行する機能を持つI/Oトランザクション発行部120、プロセッサトランザクションを処理するプロセッサトランザクション処理部130およびプロセッサトランザクションリトライフラグレジスタ140を有する。ここでプロセッサトランザクションリトライフラグレジスタ140はリトライビットというマルチプロセッサシステム内の全てのI/Oバスに個別に対応した初期値が‘0’であるエントリを持ち、何れのI/Oバスからのトランザクションがリトライされたかを記録する。

【0019】ここで、I/Oトランザクション発行部110から発行されるトランザクションのフォーマット

は、図2に示とおり、要求元のバスID、要求先のバスID、要求内容等のトランザクション自体を示すフィールド30と、ヘッダフラグフィールド31から成る。ヘッダフラグフィールド31は、順次発行するトランザクションのうち、単一のI/Oバスに接続されるデバイスからの連続する一群のトランザクションの先頭のトランザクションでは“1”、後続するトランザクションでは“0”とする。また、後で詳述するように本実施例では、単一のバスからの一連のトランザクションの転送が完了する前に要求先でリトライが発生すると、最初にリトライされたトランザクションだけでなく、一連のトランザクション中の後続するトランザクションにもリトライとされ、それらのトランザクションは再発行することになる。その再発行の先頭のトランザクションに付すべッダフラグフィールドも“1”である。I/Oトランザクション発行部110は、このように一連のトランザクションの先頭か否かで区別してヘッダフラグを付加する機能をもつ。

【0020】ノードコントローラユニット21は複数のプロセッサバスB11～B12とのプロセッサバスインターフェース200、プロセッサバスB11～B12からのトランザクションを格納するプロセッサリクエストキューリストキュー210、プロセッサリクエストキューリストキュー210に格納されたトランザクションを先行のトランザクションの完了を待たずに順次発行する機能を持つプロセッサトランザクション発行部220、I/Oトランザクションを処理するI/Oトランザクション処理部230およびI/Oトランザクションリトライフラグレジスタ240を有する。ここでプロセッサトランザクション発行部220から発行されるトランザクションも図2と同様なフォーマットを有する。すなわち、プロセッサトランザクション発行部220は单一プロセッサバスからの一連のプロセッサトランザクションの先頭か否かを区別するヘッダフラグフィールドを付加してプロセッサトランザクションを発行、または再発行する。I/Oトランザクションリトライフラグレジスタ240はリトライビットというマルチプロセッサシステム内の全てのI/Oバスに個別に対応した初期値が‘0’であるエントリを持ち、何れのI/Oバスからのトランザクションがリトライされたかを記録する。

【0021】以下一連のトランザクションTa～Tdの連続発行において、Tdでリトライが発生した場合を例に順序を保証する方法について述べる。

【0022】まず図3～図7を用いてI/Oトランザクションの順序保証について説明する。

【0023】図3はノード1aに対して、I/OバスA11の上のデバイスから一連のトランザクションTa～Tdが発行された状態を表している。I/Oリクエストキューリストキュー110の各エントリはトランザクションフィールド110aおよびトランザクション属性フィールド11

0 b で構成されており、発行されたトランザクションはトランザクションフィールド 110 a に順に格納される。トランザクション属性フィールド 110 b には各トランザクションが始めて要求されたトランザクションであること、つまり要求先のノードに一度発行したが要求先で処理できないためリトライが要求され、再度発行しようとするトランザクションではないことが記述されている。この例では具体的には '0' がセットされている。

【0024】I/Oトランザクション送信部 120 はネットワーク C11 を介して要求先のノード 1 b c に対してトランザクションを発行する。このとき I/Oトランザクション送信部 120 にて付加されるヘッダフラグフィールド 31 の値は、先頭のトランザクション T a については "1"、後続するトランザクションについては "0" である。なお、トランザクション T d の後方に、例えば I/Oバス A12 等の別の I/Oバスからの一連のトランザクションが入力すれば、その別の I/Oバスからの一連のトランザクションの先頭を I/Oトランザクション送信部 120 から発行する際に、そのヘッダフラグフィールドは "1" とされる。

【0025】図4～図5ではトランザクション受信時の振る舞いについて述べる。

【0026】まず、要求先のノードのトランザクション処理部でのアルゴリズムは次のとおりである。

【0027】①リトライフラグレジスタの受信トランザクションに対応するビットが "0" なら、トランザクション処理が実行可能か否かを判定し、可能ならOKのアクノリッジ（確認応答）を要求元に返して処理を実行する。不可能なら要求元にリトライを要求し、リトライフラグレジスタの対応ビットに "1" を記録する。

【0028】②リトライフラグレジスタの受信トランザクションに対応するビットが "1" で、受信トランザクションのヘッダフラグが "1" なら、トランザクション処理が実行可能か否かを判定する。実行可能なら、OKのアクノリッジを要求元に返してトランザクション処理を実行し、かつリトライフラグレジスタの対応するビットを "0" にクリアする。実行不可能なら、リトライ要求を要求元に返す。

【0029】③リトライフラグレジスタの受信トランザクションに対応するビットが "1" で、かつ受信トランザクションのヘッダフラグが "0" なら、トランザクション処理を実行せず、自動的にリトライ要求を要求元に返す。

【0030】図4にて、要求先のノード 1 b の I/Oトランザクション処理部 230 がトランザクション T a を受信すると、ヘッダフラグが付与されていることから T a が一連のトランザクションの先頭であることを知る。

さらに図4に示すように T a を発行した I/Oバス A1 1 に対応する I/Oトランザクションリトライフラグレ

ジスタ 240 のリトライビット 240 a が '0' であることから、T a の発生源の I/Oバスからのトランザクションはリトライ中でないことを知る。その結果 I/Oトランザクション処理部 230 はトランザクション処理が可能かを判定する。ここでは、実行可能なので、T a に対してトランザクション処理を実行し、かつノード 1 a にリトライを返す。

【0031】次に I/Oトランザクション処理部 230 がトランザクション T b を受信すると、ここでも I/Oトランザクションリトライフラグレジスタ 240 の対応するリトライビット 240 a が '0' であるため T b にリトライフラグが付与されていない（ヘッダフラグフィールドの値が "0" である）ため、処理実行可能かを判定し、実行可能なので T b に対して通常の処理を行い、かつノード 1 a にリトライを返す。

【0032】I/Oトランザクション処理部 230 が T c を受信すると、ここでも I/Oトランザクションリトライフラグレジスタ 240 の対応するリトライビット 240 a が "0" なので、実行可能かの判定を行う。しかし、ここでは何らかの理由で T c の処理が行えないと判定され、I/Oトランザクション処理部 230 は要求元のノード 1 a に対してリトライを要求する。このとき図5に示すように I/Oトランザクションリトライフラグレジスタ 240 の I/Oバス A11 に対応するリトライビット 240 a に '1' をセットする。つまり、I/Oバス A11 を要求元とするトランザクションにはリトライを要求したことが記録される。

【0033】続いて I/Oトランザクション処理部 230 が T d を受信する。I/Oトランザクションリトライフラグレジスタ 240 の対応するリトライビット 240 a が '1' であり、かつ受信した T d にヘッダフラグが付与されていないので、I/Oトランザクション処理部 230 は受信したトランザクションについてリトライを要求する。つまり I/Oトランザクション処理部 230 はトランザクション T d については処理可能か否かに関わりなくリトライを要求する。

【0034】図6～図7ではリトライ発生時の振る舞いについて述べる。

【0035】まずここで、I/Oトランザクション発行部 120 トランザクション発行のアルゴリズムを述べる。

【0036】①基本的には、I/Oリクエストキュー 110 に貯まっている I/Oバスからのトランザクション要求をキューの順番に I/Oトランザクションとして発行する（確認応答を待たない順次発行）。

【0037】②要求先からOKのアクノリッジ（確認応答）が返送されたら、それに対応するトランザクションをキューから削除する。

【0038】③要求先からリトライの要求が返送されたら、上記①の順番発行を停止し、I/Oリクエストキュー

ー110に残っている対応するトランザクションについてトランザクション属性フィールド110bに'1'をセットし、再発行する。

【0039】④上記発行に当たり、同一I/Oバスからの一連のI/Oトランザクションのもしくは先頭にはヘッダフラグを付す。上記③の再発行の場合も先の再発行に続く後続でなく再発行の先頭なら同様にヘッダフラグを付す。

【0040】⑤再発行している状態で要求先からOKのアクノリッジ(確認応答)が返送されたら、上記①の順次発行に戻る。

【0041】図の例に戻り、説明を続けると、ノード1aではTa、Tbに関してはノード1bからOKが返ってきたためI/Oリクエストキュー110からTaとTbのエントリを削除する。しかしTc、Tdに関してはノード1bからリトライ要求を受けたため、図6に示すようにトランザクション属性フィールド110bにはリトライ要求されたトランザクションであることを示す'1'がセットされる。I/Oトランザクション送信部120はリトライ要求に応じてトランザクションを再送する。このときトランザクションTcが先頭トランザクションになるためヘッダフラグ31を付与して再発行する。

【0042】ノード1bでは受信したTcにヘッダフラグが付与され、かつ図5に示すようにI/Oトランザクションリトライフラグレジスタ240の対応するリトライビット240aが'1'であるため、Tcが再発行されたトランザクションの先頭であることを知る。この時、I/Oトランザクション処理部230はトランザクション処理が可能かを確認し、可能なら図7に示すようにI/Oトランザクション処理部230はI/Oトランザクションリトライフラグレジスタ240の対応するリトライビット240aを'0'にクリアし、Tc処理を実行する。リトライビットが'0'となれば、I/Oトランザクション処理部230の処理は最初と同じとなり、Tdについても処理可能の判定を経て処理が実行される。

【0043】なお、再発行のトランザクションの先頭Tcを受信したとき、I/Oトランザクション処理部230で相変わらず処理が不可能と判定されればTcについて再びリトライが要求される。すると後続するTbについてもリトライフラグが'1'なのでリトライが要求される。処理可能かの判定がなされ、つまりリトライの状態が解消されるチャンスは、再発行中の一連のトランザクション(Tc、Td)のうちの先頭のもの(Tc)を受信した時のみである。

【0044】以上のように、1つのI/Oバス上のデバイスを発生源とする一連のI/Oトランザクションについては、要求先のノードにて一旦リトライ要求を返すと後続のトランザクションもリトライ要求を返すこととな

る。したがって、処理されないI/Oトランザクションは全て再発行されることになる。また要求元で再発行として残るトランザクションの要求先での扱い(処理かそれともリトライか)は途中では変化しない。したがって同一I/Oバスからの一連のI/Oトランザクションは、要求先ノードの状態が処理可能となったタイミングに関わらず、常に順序保証される。続いて図8～図13を用いてプロセッサトランザクションの順序保証について説明する。プロセッサトランザクションの発行に関するプロセッサトランザクション発行部220の処理アルゴリズムは、先に述べたI/Oトランザクションの発行に関するI/Oトランザクション発行部120の処理アルゴリズムと同様である。また、プロセッサトランザクションの要求先のプロセッサトランザクション処理部130の処理アルゴリズムもI/Oトランザクションの処理に関するI/Oトランザクション処理部の処理アルゴリズムと同様である。

【0045】順にのべると、図8はノード1aに対してあるプロセッサユニットからトランザクションTa～Tdが発行された状態を表している。プロセッサリクエストキュー210の各エントリもトランザクションフィールド210aおよびトランザクション属性フィールド210bで構成されており、発行されたトランザクションはトランザクションフィールド210aに順に格納される。トランザクション属性フィールド210bには各トランザクションが始めて要求されたトランザクションであるため'0'がセットされている。

【0046】続いてプロセッサトランザクション発行部220はネットワークC11を介して要求先のノード1bに對してトランザクションを発行する。ここでもI/Oトランザクションと同様に各トランザクションに対して図2に示すようなヘッダフラグフィールドが付与される。プロセッサトランザクション発行部220はTaが一連のトランザクションの先頭であるため、ヘッダフラグフィールドに'1'をセットする。後続のトランザクションのヘッダフィールドには'0'をセットする。

【0047】図9～図10ではトランザクション受信時の振る舞いについて述べる。

【0048】要求先のノード1bではプロセッサトランザクション処理部130において受信したトランザクションが順次処理される。プロセッサトランザクション処理部130がトランザクションTaを受信すると、図9に示すようにTaを発行したプロセッサバスに対応するプロセッサトランザクションリトライフラグレジスタ140の対応するリトライビット140aが'0'であることからTaがはじめて発行されたトランザクションであることを知る。その結果プロセッサトランザクション処理部130はTaの処理可能かを判定し、ここでは処理可能であるのでトランザクション処理を実行し、ノード1aに對してOKを返す。

【0049】次にプロセッサトランザクション処理部130がトランザクションTbを受信すると、ここでも対応するリトライビット140aが‘0’であるため処理可能かの判定を行う。処理可能なため、Tbに対して通常の処理を行い、かつノード1aに對してOKを返す。

【0050】プロセッサトランザクション処理部130がTcを受信すると、対応するリトライビット140aが‘0’なので処理可能かの判定がおこなわれるが、ここでは何らかの理由でTcの処理が行えないので、プロセッサトランザクション処理部130は要求元のノード1aに對してリトライを要求する。このとき図10に示すようにプロセッサトランザクションリトライフラグレジスタ140の対応するリトライビット140aに‘1’をセットする。続いてプロセッサトランザクション処理部130がTdを受信すると、プロセッサトランザクション処理部130の対応するリトライビット140aが‘1’で、かつTcにヘッダフラグが付されていないのでプロセッサトランザクション処理部130はTdに關してもリトライを要求する。

【0051】図11～図12ではリトライ発生時の振る舞いについて述べる。ノード1aではTa、Tbに關してはノード1bからOKが返ってきたためプロセッサリクエストキュー210からTaとTbのエントリを削除する。しかしTc、Tdに關してはノード1bからリトライ要求を受けたため、図11に示すようにトランザクション属性フィールド210bにはリトライ要求されたトランザクションであることを表す値である‘1’をセットする。プロセッサトランザクション発行部220はリトライ要求に応じてトランザクションを再送する。このときトランザクションTcが先頭トランザクションになるためヘッダフラグを付与して再発行する。

【0052】ノード1bでは受信したTcにヘッダフラグが付与され、かつ図10に示すようにプロセッサトランザクションリトライフラグレジスタ140の対応するリトライビット140aが‘1’であるため、Tcが再発行されたトランザクションの先頭であることを知る。その結果トランザクション処理が可能かを判定し、ここでは処理可能であるため、図12に示すようにプロセッサトランザクション処理部130はプロセッサトランザクションリトライフラグレジスタ140の対応するリトライビット140aを‘0’にクリアし、Tcの処理を実行する。ついで、Tdを受信した際には、既に対応するリトライビット140aが‘0’であるので、Tdに關しては先のTa、Tbと同じ扱いで処理がなされる。

【0053】以上の処理により、あるプロセッサバスからのトランザクションがリトライされた場合においても、I/Oトランザクションのリトライと同様に後続のトランザクションもリトライすることができ、トランザクションの順序が保証される。

【0054】以上にのべた実施例では、一連のトランザクションの1つについてリトライの要求が返送されると、要求元は対応するトランザクションを再発行し、以降は要求元は特殊な再発行のモードとなる。つまり、確認応答にかかわりない順次発行の動作でなく、次のリトライ要求を待つて次を再発行す動作となる。これに代え、一連のトランザクションの1つについてリトライの要求が返送された場合に、同一バスを発生源とする再発行すべきトランザクションの一連の再発行を、個々の応答を待たずに順次行う構成とすることも可能である。この変形例における要求元のトランザクション発行部のアルゴリズムは以下の通りである。

【0055】①リクエストキューに貯まっているバスからのトランザクションを、キューの順番で発行する。

【0056】②要求先からOKのアクノリッジ（確認応答）が返送されたら、それに対応するトランザクションをキューから削除する。（以上は上記実施例と同一）

③要求先からリトライの要求が返送されたら、対応するトランザクションだけでなく、リクエストキューに残っていて既に発行済みの、同一バスを発生源と一連のトランザクションの全てについてトランザクション属性フィールドに‘1’をセットする。そして、属性フィールドが‘1’の一連のトランザクションを、先頭のも（リトライ要求に対応するもの）のから順に連続して再発行する。

【0057】④上記発行に当たり、同一バスからの一連のトランザクションの先頭にはヘッダフラグを付す。上記③の再発行の場合も、先の再発行に続く後続ではなくて、再発行の先頭なら同様にヘッダフラグを付す。（これについても上記実施例と同じ）

⑤再発行している状態で要求先からOKのアクノリッジ（確認応答）が返送されたら、対応するトランザクションをリクエストキューから削除するとともに、それに後続する同一バスからのトランザクションのトランザクション属性フィールドを‘1’から‘0’にリセットする。

【0058】上記変形例においては、リトライとされたトランザクションの後続トランザクションは自動的に要求元から再発行される。したがって、それら後続トランザクションの個々についてリトライの要求を要求元から要求元に返送する必要はなく、その後続トランザクションを受信してもトランザクション処理を行わなければ良い。要求先となるトランザクション処理部の処理アルゴリズムは、以下の通りとなる。

【0059】①リトライフラグレジスタの受信トランザクションに対応するビットが“0”なら、トランザクション処理が実行可能か否かを判定し、可能ならOKのアクノリッジ（確認応答）を要求元に返して処理を実行する。不可能なら要求元にリトライを要求し、リトライフラグレジスタの対応ビットに“1”を記録する。（この点は最初の実施例と同じ）

②リトライフラグレジスタの受信トランザクションに対応するビットが"1"で、受信トランザクションのヘッダフラグが"1"なら、つまり再発行される一連トランザクションの先頭を受信したか、もしくは一連の再発行の後の別のバスを発生源とするトランザクションの先頭を受信した場合には、処理が実行可能か否かを判定する。実行可能なら、OKのアクノリッジを要求元に返してトランザクション処理を実行し、かつリトライフラグレジスタの対応するビットを"0"にクリアする。実行不可能なら、リトライ要求を要求元に返す。

【0060】③リトライフラグレジスタの受信トランザクションに対応するビットが"1"で、かつ受信トランザクションのヘッダフラグが"0"なら、トランザクション処理が実行可能か否かの判定を行わず、トランザクション処理も行わない。

【0061】以上のような変形例では、一連の再発行が終わっても、再発行の先頭についてのOKの確認応答もしくはリトライの要求が返らない場合があり得る。この時、リクエストキューに再発行トランザクションの後ろに、さらに同一バスからのトランザクションが貯まっているれば、それらを続けて発行しても良い。また、別のバスを発生源とする別の一連のトランザクションが貯まっているれば、上記要求側アルゴリズムの④により、その先頭にヘッダフラグをつけて発行を行う。いずれの場合も、もし後から再発行トランザクションの先頭について再度リトライが要求されたら、そのときは上記要求側アルゴリズムの③により、改めて同一バスからの発行済みのトランザクション全てを再発行とするので、処理順序は保証される。

【0062】以上のように、リトライ後の再発行を連続発行とした変形例においても、同一バスからの一連のトランザクションについて処理の順序保証が可能となる。要求側のトランザクション発行部において、リクエストキューのトランザクション属性フィールドを管理するためのハードウェア量は先の実施例より大きくなるが、要求先からの応答のオーバーヘッドが小さく、一連の再発行においても要求先からの応答に関わりない連続発行が可能なので、より高速のトランザクション処理が実現できる。

【0063】また、以上の実施態様では、トランザクションの管理単位が個々のI/Oバス単位、プロセッサバス単位であった。これに代え、管理単位を個々のI/Oデバイス単位、プロセッサユニット単位とする構成も可能である。プロセッサバス単位からプロセッサユニット単位に代えた構成について述べると以下のようになる。まず、図1のノードコントローラユニット21では、受け付けるプロセッサトランザクションについて、個々のプロセッサユニット25、26等を発生源として認識する。プロセッサトランザクションの要求先である各I/Oユニットのリトライフラグレジスタ140には個別の

プロセッサユニットに対応するエントリーが設けられる。要求先I/Oユニットのプロセッサトランザクション処理部140で、あるプロセッサユニットを発生源とするプロセッサトランザクションをリトライしたなら、そのプロセッサユニットに対応するエントリーにリトライフラグを記入し、これを用いて同一のプロセッサユニットを発生源とする後続するプロセッサトランザクションについてもリトライする。

【0064】なお実施形態のマルチプロセッサシステム10ではトランザクションの要求先のノードがリトライ制御を行っているが、実施形態の変形例として、コヒーレント制御されたトランザクションを全ノードがスヌープしている場合においては任意のノードがリトライ制御を行うことが可能である。

【0065】

【発明の効果】以上説明したように、本発明によれば以下のようない効果がある。

【0066】先行のトランザクションの完了を待たずして後続のトランザクションを連続発行することができるため、システム全体のスループットおよび個々のトランザクションが終了するまでのレイテンシを向上させることができる。

【図面の簡単な説明】

【図1】図1は本発明を実現するマルチプロセッサシステムの構成を示す概略図である。

【図2】トランザクションに付与されるヘッダフラグフィールドを説明するための概略図である。

【図3】I/Oユニット11がI/Oトランザクションを発行する方法を説明するための模式図である。

30 【図4】ノードコントローラユニット21がI/Oトランザクションを受信した際の動作を説明するための模式図である。

【図5】ノードコントローラユニット21がI/Oトランザクションをリトライした際の動作を説明するための模式図である。

【図6】I/Oユニット11がリトライ要求されたI/Oトランザクションを再発行する際の動作を説明するための模式図である。

40 【図7】ノードコントローラユニット21が再発行されたI/Oトランザクションを処理する際の動作を説明するための模式図である。

【図8】ノードコントローラユニット21がプロセッサトランザクションを発行する方法を説明するための模式図である。

【図9】I/Oユニット11がプロセッサトランザクションを受信した際の動作を説明するための模式図である。

【図10】I/Oユニット11がプロセッサトランザクションをリトライした際の動作を説明するための模式図50である。

【図11】ノードコントローラユニット21がリトライ要求されたプロセッサトランザクションを再発行する際の動作を説明するための模式図である。

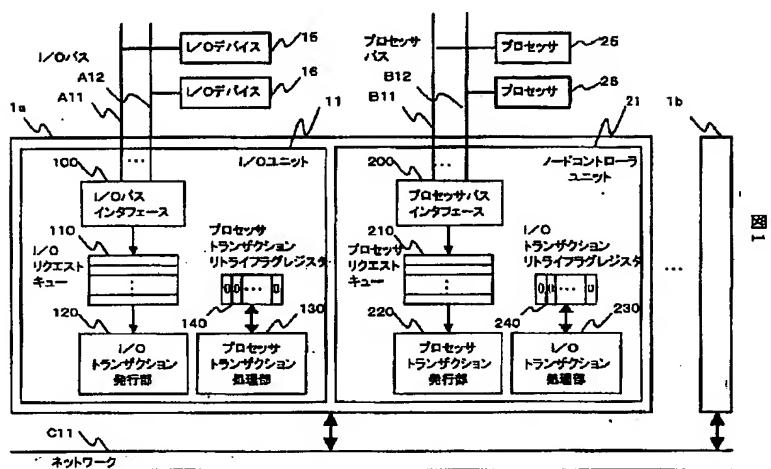
【図12】I/Oユニット11が再発行されたプロセッサトランザクションを処理する際の動作を説明するための模式図である。

【符号の説明】

1…ノード、1a…ノード、1b…ノード、11…I/Oユニット、21…ノードコントローラユニット、30…トランザクション、31…ヘッダフラグフィールド、100…I/Oバスインターフェース、110…I/Oリクエストキュー、110a…トランザクションフィールド、110b…トランザクション属性フィールド、12…\*

\* 0…I/Oトランザクション発行部、130…プロセッサトランザクション処理部、140…プロセッサトランザクションリトライフラグレジスタ、140a…リトライビット、200…プロセッサバスインターフェース、210…プロセッサリクエストキュー、210a…トランザクションフィールド、210b…トランザクション属性フィールド、220…プロセッサトランザクション発行部220、230…I/Oトランザクション処理部、240…I/Oトランザクションリトライフラグレジスタ、240a…リトライビット、A11～A12…I/Oバス、B11～B12…プロセッサバス、C11…ネットワーク。

【図1】



【図2】

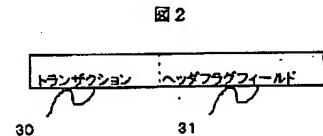
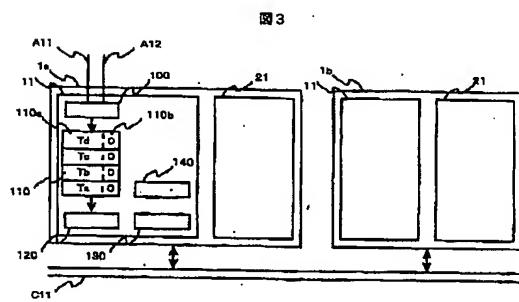
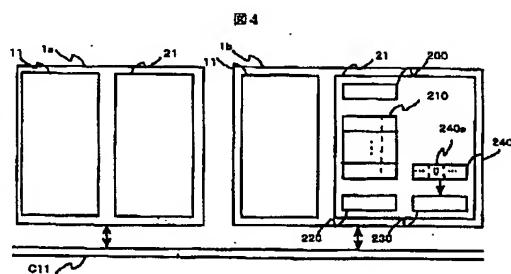


図2

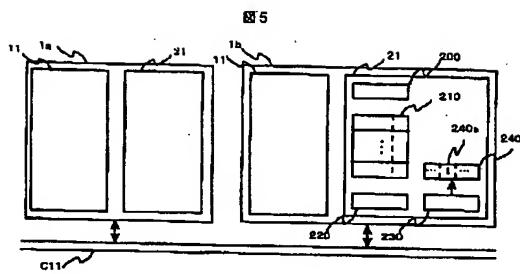
【図3】



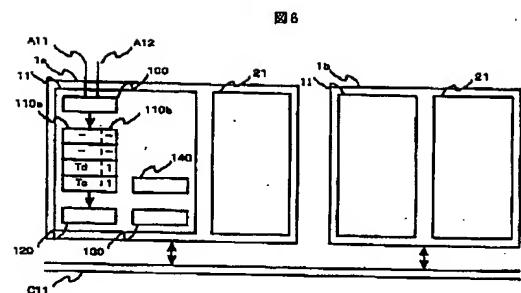
【図4】



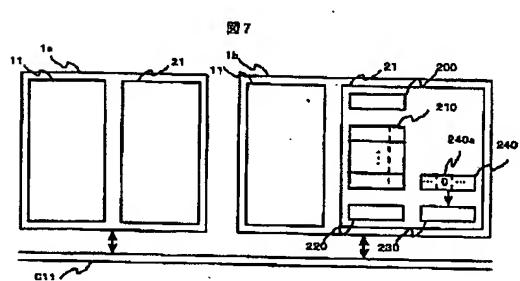
【図5】



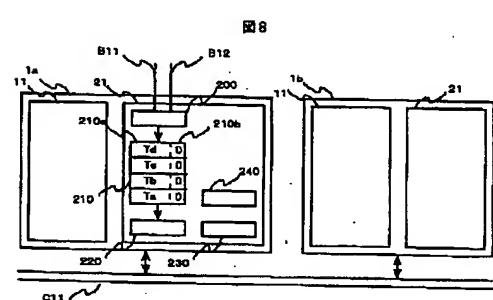
【図6】



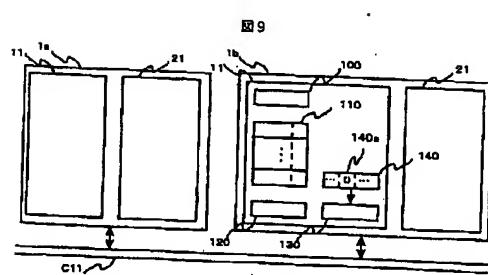
【図7】



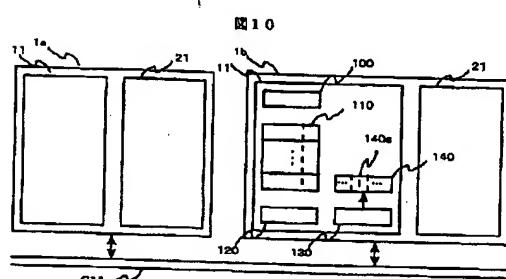
【図8】



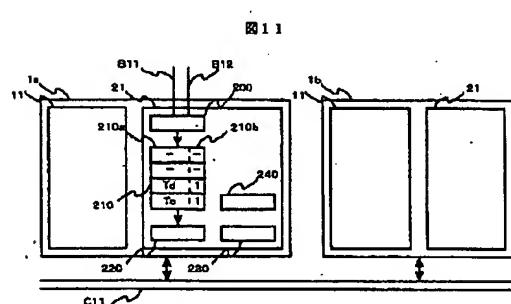
【図9】



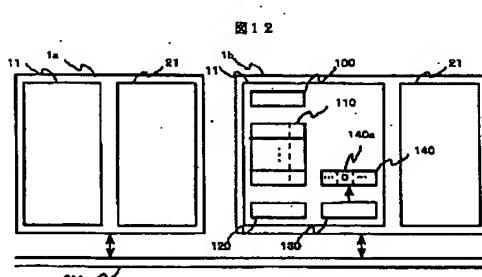
【図10】



【図11】



【図12】



## フロントページの続き

(72)発明者 上原 敬太郎  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 尾馬 雄次  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内

(72)発明者 濱中 直樹  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
F ターム(参考) 5B045 BB12 BB17 BB42  
5B077 AA23 AA33 AA34 FF06